AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method of performing embossed-style bump mapping within a pipelined graphics system using conventional RGB textures comprising:

providing object space ~~a vertex attribute descriptor that describes~~ Tangent and Binormal vector data for each of plural vertices of a polygon;

transforming Tangent and Binormal vectors to eye-space;

computing an eye-space ~~a~~ light direction vector and a normalized light direction vector;

using dedicated hardware circuitry within the graphics pipeline to compute ~~computing~~ texture coordinate displacements for each of said vertices as a function of ~~in response to~~ said light direction vector and said normalized light direction vector and said Tangent and Binormal vector data;

retrieving ~~generating~~ texture values from a texture memory corresponding to said vertices and ~~coordinate values in response~~ to said computed texture coordinate displacements; and

texture mapping said polygon based on said texture ~~coordinate~~ values to provide an embossing effect;

wherein an embossed texture effect is developed from conventional RGB texture data stored in a texture memory minus an offset texture defined by the computed texture coordinate displacement values.

~~wherein the texture coordinate displacements computing step does not use Normal vector data to compute said texture coordinate displacements.~~

2. (Currently Amended) A method as in claim 1 wherein the texture coordinate displacements are computed by hardware within the graphics pipeline without using Normal vector data. ~~where said per-vertex data includes at least a Normal, a Tangent, and a Binormal vector.~~

Claim 3 (Canceled).

4. (Currently Amended) A method as in claim 1 wherein the step of using hardware circuitry within the graphics pipeline to compute texture coordinate displacements ~~displacement computing step~~ includes performing vector dot-product computations between the computed light direction vector and the Tangent and Binormal vectors.

Claim 5 (Canceled).

6. (Currently Amended) A method of performing embossed-style bump mapping comprising:

decoding a ~~generalized~~ texture coordinate generation instruction ~~function~~ that specifies generating embossed-style bump-mapping texture displacements;

converting object oriented Tangent and Binormal vectors per vertex to eye-space;

using hardware circuitry within a graphics system to compute ~~perturbing input~~ texture offset coordinates per vertex based on eye-space binormals and light direction information in response to the decoding step;

adding ~~using~~ the per vertex computed ~~input~~ texture offset coordinates to eye-space transformed ~~and the perturbed~~ texture coordinates to obtain displaced texture coordinates for looking ~~look~~-up texels in a texture image ~~height-field bump~~ map; and

~~computing bump height information based on the texel values; and~~

combining texel information obtained using displaced texture coordinates with texel information obtained using non-displaced texture coordinates ~~the bump height information with pixel color value information~~ to provide embossed-style bump mapping;

wherein the combining step includes combining texel information ~~is performed~~ in texture processing hardware to produce an embossing effect in a displayed image.

Claim 7 (Canceled).

8. (Currently Amended) The method of claim 6 wherein the graphics system hardware circuitry is used to compute the ~~perturbing step comprises computing~~ texture coordinate offset ~~displacement~~ values based a Tangent vector and a Binormal vector and a dot-product of each vector with a light direction vector.

9. (Currently Amended) The method of claim 6 wherein the combining texel information ~~computing~~ step comprises subtracting texel data acquired using the displaced ~~perturbed~~ texture coordinates from texel data acquired using ~~input~~ texture coordinates associated with a vertex.

10. (Original) The method of claim 6 wherein the decoding step comprises decoding a generalized vertex attribute description function that specifies a Tangent vector and a Binormal vector.

11. (Original) The method of claim 10 wherein the Tangent and Binormal vectors are specified by reference to separate memory indexes.

12. (Currently Amended) In a graphics system including a processor and a separate graphics processing pipeline having vertex transformation and lighting

circuitry, the pipeline performing emboss-style bump-mapping based on ~~texels in response to~~ texture coordinate displacements computed from Tangents and Binormals, an improvement comprising:

texture coordinate displacement computation circuitry for performing embossment-style bump-mapping included within the graphics pipeline vertex transformation and lighting circuitry.

13. (Currently Amended) In a graphics system including a processor and a separate graphics processing pipeline, the pipeline performing emboss-style bump-mapping based on texture ~~texels in response to textures~~ coordinate displacements computed from Tangents and Binormals, an improvement comprising performing a ~~the~~ texture coordinate displacement computation using hardware within the pipeline.

14. (Withdrawn) A graphics processing system having vertex transformation and lighting processing hardware and an enhanced API vertex attribute description command function for specifying at least Tangent and Binormal object-space surface vectors, wherein the geometry processing and lighting hardware transforms the object-space Tangent and Binormal vectors to eye-space, computes an eye-space light direction vector based a light position and a vertex position, and performs vector dot-product computations between the computed light direction vector and the transformed Tangent and Binormal surface vectors to generate texture coordinate displacements for

use in creating an embossed texture effect, and wherein the Tangent and Binormal vectors are scaled by scaling a model view matrix and applying the scaled model view matrix to the Tangent and Binormal vectors.

15. (Currently Amended) In a graphics processing system that renders and displays images at least in part in response to polygon vertex attribute data and texture color data stored in an associated memory, a vertex transformation and lighting processing portion embodied in hardware, comprising:

a vector processing unit comprising at least two distinct dot-product computation circuits for computing vector dot-products; and

circuitry connected to said vector processing unit ~~a bump-mapping unit~~ for computing at least a ~~normalized~~ light-to-vertex direction vector and computing a set of texture displacement values from Tangent, Binormal and light-to-vertex direction vector data for use in creating an embossed texture effect.

16. (Currently Amended) In a graphics processing system that renders and displays images at least in part in response to polygon vertex attribute data and texture color data stored in an associated memory, vertex transformation and lighting processing hardware comprising:

vector processing circuitry comprising a plurality of dot-product computation units; and

texture coordinate offset computation ~~bump-mapping~~ circuitry comprising an inverse square-root computation circuit for computing a reciprocal of a square-root of an input value and at least one floating point multiplier and one floating point adder;

wherein said vector processing circuitry and said texture coordinate offset computation circuitry is used to compute texture coordinate displacement values for generating emboss-style bump mapping effects in a displayed image.

17. (Currently Amended) The geometry vertex and lighting processing hardware of claim 16 wherein the texture coordinate offset computation circuitry ~~bump-mapping unit~~ further comprises a FIFO input buffer for storing incoming texture coordinate values.

18. (Currently Amended) The vertex transformation and lighting processing hardware of claim 16 wherein the texture coordinate offset computation circuitry ~~bump-mapping unit~~ further comprises a floating point adder for computing a light-to-vertex vector.

19. (Original) The vertex transformation and lighting processing hardware of claim 16 wherein the dot-product computation units comprise at least one floating point multiplier and one floating point adder.

20. (Currently Amended) In a graphics processing system that renders and displays images at least in part in response to polygon vertex attribute data and texture color data stored in an associated memory, the graphics system including vertex transformation and lighting processing hardware for generating texture coordinate displacement values for implementing at least embossed-style bump-mapped texture effects, the vertex transformation and lighting processing hardware comprising:

a first vector dot-product computation unit for transforming vector data to eye-space;

a second vector dot-product computation unit for computing lighting direction vector dot-products;

an inverse square-root computation unit for computing a reciprocal of a square-root of an input value;

at least one floating point multiplier unit; and

at least one floating point adder unit;

wherein in response to an API <u>embossed-style</u> bump mapping function instruction the vertex transformation and lighting processing hardware transforms <u>object space Tangent and Binormal</u> vector data per-vertex into eye-space, computes texture coordinate displacement values based on lighting direction vector dot-products, and adds the displacement values to texture coordinates for use in producing emboss-style bump-mapped texture effects.

21. (Original) The graphics processing system of claim 20 further including a graphics API vertex attribute function which specifies at least Normal, Tangent and Binormal vectors per vertex, or specifies an index to at least each of these vectors stored in memory.

22. (Currently Amended) The graphics processing system of claim 20 wherein the API <u>embossed-style</u> bump mapping function is defined to specify one of at least eight different textures for producing embossing effects.

23. (Original) The graphics processing system of claim 20 wherein the Normal, Tangent and Binormal vectors each comprise three 32-bit vector elements.

24. (Withdrawn) In a graphics processing system that renders and displays images at least in part in response to polygon vertex attribute data and texture color data stored in an associated memory, the graphics system including a geometry transform unit comprising hardware for at least computing a coordinate-space transformation and a vector dot-product, a method of implementing embossed-style bump-mapped texture effects in graphics rendering system, comprising the steps of:

storing a texture data image in memory, the texture data image comprising color values parameterized by at least two coordinate values representing two orthogonal axes for mapping the image;

supplying light position information, texture coordinate information, vertex position information and object-space Normal, Binormal and Tangent vector data per polygon vertex to the geometry transform unit, wherein for each vertex said Binormal and Tangent vector data map respectively, in an object-space coordinate system, to each orthogonal axis of the bump-map image;

transforming the object-space Normal, Binormal and Tangent vector data to an eye-space coordinate system;

computing a light direction vector from light position and vertex position information;

computing a texture coordinate displacement based on a vector dot-product between the light direction vector and each of the Binormal and Tangent eye-space vector components;

adding the texture coordinate displacement to eye-space texture coordinates to obtain a set of displaced texture coordinates;

using the set of displaced texture coordinates to retrieve texture color data from the stored texture data image; and

performing texture subtraction in one pass.

25. (Withdrawn) A method of performing embossed-style bump mapping comprising:

providing a description of Tangent and Binormal vectors for each of plural vertices of a polygon;

providing a light direction vector;

computing texture coordinate displacements for each of said vertices in response to said light director vector and said Tangent and Binormal vector;

generating texture coordinates in response to said computed texture coordinated displacements; and

texture mapping said polygon based on said texture coordinates, including providing a texture combining operation that performs texture subtraction in a single pass.

26. (Withdrawn) The method of claim 25 wherein said texture combining is performed in texture hardware.

27. (Withdrawn) The method of claim 25 further including scaling the Tangent and Binormal vector data by scaling a model view matrix and applying the scaled model view matrix to the vector data.

28. (Withdrawn) The method of claim 25 wherein the texture coordinate displacement computing does not use a Normal vector.

29. (Withdrawn) The method of claim 25 wherein the texture coordinate displacement computing computes the following in parallel:

a first vector dot-product between the light direction vector and the Tangent vector,

a second vector dot-product between the light direction vector and the Binormal vector, and

the square of the light direction vector.

30. (Withdrawn) The method as in claim 25 wherein the texture coordinate displacement computing step is performed using two distinct dot-product computation units, the first dot-product computation unit computing eye-space transformation of the Tangent and Binormal vectors, the second dot-product computation unit computing at least vector dot-products between the light direction vector and each of the Tangent and Binormal vectors.

31. (Withdrawn) In a graphics chip including a logic array, a pipelined arrangement implemented within the logic array that performs embossed-style bump mapping based on Tangent and Binormal vectors for each of plural vertices of a polygon and a light direction vector, said arrangement including:

a dot-product computation unit and associated logic circuitry adapted to receive a scaling factor, the dot-product computation unit and associated logic circuitry scaling a model view matrix in response to the scaling factor and applying the scaled model view matrix to the Tangent and Binormal vectors to provide texture coordinate displacements for each of said vertices; and

texture mapping and combining circuitry that generates embossing effects in response to said texture coordinate displacements.

32. (Withdrawn) Apparatus as in claim 31 wherein said texture mapping and combining circuitry performs texture subtraction in one pass.

33. (Withdrawn) Apparatus as in claim 31 wherein the dot-product computation unit does not use the Normal input vector to compute displacements.

34. (Withdrawn) Apparatus as in claim 31 further including a further dot-product computation unit that parallelly computes dot products between the Binormal and Tangent vectors and a light direction vector.